INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2000; **00**:1–6 *Prepared using* fldauth.cls *[Version: 2002/09/18 v1.01]*

High-order semi-implicit time-integration of a triangular discontinuous Galerkin oceanic shallow water model

F.X. Giraldo†*and M. Restelli‡

†*Department of Applied Mathematics, Naval Postgraduate School, Monterey, CA 93943*
‡*Max-Planck Institute, Hamburg Germany*

SUMMARY

We extend the explicit in time high-order triangular discontinuous Galerkin (DG) method to semi-implicit and then apply the algorithm to the two-dimensional oceanic shallow water equations; we implement high-order semi-implicit time-integrators using the backward difference formulas from orders one through six. The reason for changing the time-integration method from explicit to semi-implicit is that the explicit method requires a very small time-step in order to maintain stability, especially for high-order DG methods. Changing the time-integration method to be semi-implicit allows one to circumvent the stability criterion due to the gravity waves, which for most shallow water applications are the fastest waves in the system (the exception being supercritical flow where the Froude number is greater than one). The challenge of constructing a semi-implicit method for a DG model is that the DG machinery requires not only the standard finite element-type (FE) area integrals but finite volume-type (FV) boundary integrals as well. These boundary integrals pose the biggest challenge in a semi-implicit discretization because they require the construction of a Riemann solver that is the true linear representation of the nonlinear Riemann problem; if this condition is not satisfied then the resulting numerical method will not be consistent with the continuous equations. In this paper we present semi-implicit time-integrators for the DG method that maintain most of the usual attributes associated with DG methods such as: high-order accuracy (in both space and time), parallel efficiency, excellent stability, and conservation. The only property lost is that of a compact communication stencil typical of time-explicit DG methods; implicit methods will always require a much larger communication stencil. We apply the new high-order semi-implicit discontinuous Galerkin method to the shallow water equations and show results for many standard test cases of oceanic interest such as: standing, Kelvin and Rossby soliton waves, and the Stommel problem. The results show that the new high-order semi-implicit DG model, that has already been shown to yield exponentially convergent solutions for smooth problems, results in a more efficient model than its explicit counterpart. Furthermore, the capacity to use high-order time-integrators offers a big advantage in accuracy when simulating time-dependent problems especially when using high-order DG methods; without high-order time-integration it makes little sense to use high-order spatial discretizations.

KEY WORDS: discontinuous Galerkin; explicit; finite element; finite volume; implicit; semi-implicit; shallow water; triangle.

*Correspondence to: fxgirald@nps.edu

Received September 2008
Revised January 2007

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2000		**00-00-2000 to 00-00-2000**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
High-order semi-implicit time-integration of a triangular discontinuous Galerkin oceanic shallow water model	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Postgraduate School,Department of Applied Mathematics,Monterey,CA,93943**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**24**	

1. Introduction

The discontinuous Galerkin (DG) method has come into prominence in the last decade in all areas of numerical modeling; however, it has only been in the last few years that this method has received attention in geophysical fluid dynamics. The high-order accuracy, geometric flexibility to use unstructured grids, conservation, and monotonicity properties of the DG method makes it a prime candidate for the construction of future ocean and shallow water models. The advantages offered by the DG method will benefit all areas of ocean modeling but, specifically, it will improve coastal ocean models where proper coastline representation, and the ability to handle steep gradients should translate into better modeling of tsunamis, storm surges, and hurricanes. Let us now review the literature concerning the application of the DG method to the oceanic shallow water equations.

Schwaneberg and Köngeter (2000) [32] first used the DG method for the planar shallow water equations, followed by the work of Li and Liu (2001) [26], and Aizinger and Dawson (2002) [2]. Dupont and Lin (2004) [12], Eskilsson and Sherwin (2004) [13], Remacle et al. (2006) [28], and Kubatko et al. (2006) [25] constructed shallow water models on triangles using a collapsed local coordinate discontinuous Galerkin method. Giraldo and Warburton (2008) [20] developed a high-order DG oceanic shallow water model on unstructured adaptive triangular grids. In that paper, we showed that the model yields exponentially convergent solutions (for smooth problems). However, we used explicit time-integration methods which, while easy to implement, require small time-steps in order to maintain stability. To ameliorate this deficiency found in all DG shallow water models, we extend the explicit time-integrators to semi-implicit. To date, there has been no work on the development of semi-implicit time-integrators for shallow water DG models; all of the DG shallow water models found in the literature use explicit time-stepping, including those discussed above. Furthermore, the only work on DG and semi-implicit methods found in the literature are the papers by Dolejsi, Feistauer, and co-authors on the compressible Navier-Stokes equations (see [11], [10], [14], [15], [9]). Their semi-implicit DG formulation is based on low-order polynomial spaces (third order or less) and their approach is fundamentally different from ours in that they rely on a linearization of the nonlinear operators in conjunction with a special flux function that facilitates this linearization. Our approach [29] relies on extracting the linear operators containing the fastest wave speeds in the system and then discretizing them implicitly in time. While both approaches are very effective, our approach is more similar to the classical semi-implicit method first proposed by Robert et al. (1972) [31]. The advantage of this approach is that, once the numerical machinery is developed, it can be applied to any equation set with minimal modifications. Moreover, the semi-implicit DG approach is easily extendable to generalized families of linear multi-step time-integration methods as we show here.

The remainder of the paper is organized as follows. Section 2 describes the governing equations of motion used to test our numerical method. In Sec. 3 we describe the spatial discretization of the governing equations and in Sec. 4 the time-integrators used. Finally, in Sec. 5 we present comparisons between the explicit and semi-implicit versions of the model. This then leads to a summary on the direction of future work.

2. Continuous Equations

The oceanic shallow water equations are a system of nonlinear partial differential equations which govern the motion of a viscous incompressible fluid in a shallow depth. The predominant feature of this type of fluid is that the characteristic length of the fluid is far greater than its depth, this is analogous to ocean flow problems and is the reason these equations are typically used as a first step toward the construction of ocean models.

The shallow water equations in conservation form are

$$\frac{\partial \boldsymbol{q}}{\partial t} + \boldsymbol{\nabla} \cdot \boldsymbol{F}(\boldsymbol{q}) = S(\boldsymbol{q}) \tag{1}$$

where $\boldsymbol{q} = (\phi, \boldsymbol{U}^{\mathcal{T}})^{\mathcal{T}}$ are the conservation variables,

$$\boldsymbol{F}(\boldsymbol{q}) = \begin{pmatrix} \boldsymbol{U} \\ \frac{\boldsymbol{U} \otimes \boldsymbol{U}}{\phi} + \frac{1}{2}(\phi^2 - \phi_B^2)\boldsymbol{\mathcal{I}}_2 \end{pmatrix} \tag{2}$$

is the flux tensor and

$$S(\boldsymbol{q}) = -\begin{pmatrix} 0 \\ f\,(\boldsymbol{k} \times \boldsymbol{U}) - \phi_S \boldsymbol{\nabla} \phi_B - \frac{\boldsymbol{\tau}}{\rho H} + \gamma \boldsymbol{U} \end{pmatrix} \tag{3}$$

is the source function where the nabla operator is defined as $\boldsymbol{\nabla} = (\partial_x, \partial_y)^{\mathcal{T}}$, $\otimes$ denotes the tensor product operator, $\phi = g(h_S + h_B)$ is the geopotential height where g is the gravitational constant, h_S is the free surface height of the fluid, $\phi_B = g h_B$ is the bathymetry (e.g., bottom of the ocean) which we assume constant, $\boldsymbol{U} = \phi \boldsymbol{u}$ is the momentum, $\boldsymbol{u} = (u, v)^{\mathcal{T}}$ is the velocity vector, $f = f_0 + \beta(y - y_m)$ is the Coriolis parameter, $k = (0, 0, 1)^{\mathcal{T}}$ is the unit normal vector of the x-y plane, and the term $\boldsymbol{\mathcal{I}}_2$ is a rank-2 identity matrix. The vector $\boldsymbol{\tau}$ is the wind stress, ρ is the density, H is a mean height, and the constant γ is the bottom friction.

2.1. Linearized Continuous Equations

Let us now decompose Eqs. (1) - (3) into their linear and nonlinear components. Splitting the geopotential height ϕ into the depth from mean sea level to the ocean bottom ϕ_B and the height from mean sea level to the water surface ϕ_S we then have $\phi(\boldsymbol{x}, t) = \phi_S(\boldsymbol{x}, t) + \phi_B(\boldsymbol{x})$ which we can now use to substitute into the equations to get

$$\frac{\partial \phi_S}{\partial t} + \boldsymbol{\nabla} \cdot \boldsymbol{U} = 0 \tag{4}$$

$$\begin{aligned} \frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{\nabla} \cdot \left[\delta_{NL} \left(\frac{\boldsymbol{U} \otimes \boldsymbol{U}}{\phi} + + \frac{1}{2} \phi_S^2 \boldsymbol{\mathcal{I}}_2 \right) + \phi_S \phi_B \boldsymbol{\mathcal{I}}_2 \right] &= -f(\boldsymbol{k} \times \boldsymbol{U}) + \phi_S \boldsymbol{\nabla} \phi_B \\ &+ \frac{\boldsymbol{\tau}}{\rho H} - \gamma \boldsymbol{U} \end{aligned} \tag{5}$$

where δ_{NL} is a switch which retains the nonlinear terms when $\delta_{NL} = 1$ and turns them off for $\delta_{NL} = 0$.

In the next few sections, we describe the semi-implicit (SI) method for the oceanic shallow water equations in the particular case that the discontinuous Galerkin method is used to represent spatial derivatives. As the reader will see, a pivotal component of the SI method is

the construction of a linearized form of the continuous equations. Linearizing Eqs. (1) - (3) yields

$$\begin{aligned} \frac{\partial \phi_S}{\partial t} + \boldsymbol{\nabla} \cdot \boldsymbol{U} &= 0 \\ \frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{\nabla} \cdot (\phi_S \phi_B \boldsymbol{\mathcal{I}}_2) &= -f(\boldsymbol{k} \times \boldsymbol{U}) + \phi_S \boldsymbol{\nabla} \phi_B + \frac{\boldsymbol{\tau}}{\rho H} - \gamma \boldsymbol{U} \end{aligned} \tag{6}$$

which are obtained by setting $\delta_{NL} = 0$ in Eqs. (4) and (5). The maximum eigenvalue of the linear system given in Eq. (6) is $\lambda_L = \sqrt{\phi_B}$ which is in fact the linearized eigenvalue of λ_{NL} obtained for the nonlinear system given in Eqs. (1) - (3). From Eq. (6) we can define the linear operator as follows

$$L = -\begin{pmatrix} \boldsymbol{\nabla} \cdot \boldsymbol{U} \\ \boldsymbol{\nabla} \cdot (\phi_S \phi_B \boldsymbol{\mathcal{I}}_2) + f(\boldsymbol{k} \times \boldsymbol{U}) - \phi_S \boldsymbol{\nabla} \phi_B + \gamma \boldsymbol{U} \end{pmatrix}. \tag{7}$$

We will return to this linear operator in Sec. 4. Let us now describe the approximation of the spatial derivatives by the DG method. We need to know this information before we can construct the semi-implicit solution.

3. Triangular Discontinuous Galerkin Method

In this section we describe the approximation of the spatial derivatives of the shallow water equations using the discontinuous Galerkin method on triangles. This includes: the choice of basis functions, integration, construction of the semi-discrete problem and its corresponding matrix form.

3.1. Basis Functions

To define the discrete local operators we begin by decomposing the domain Ω into N_e conforming non-overlapping triangular elements Ω_e such that

$$\Omega = \bigcup_{\mathrm{e}=1}^{N_\mathrm{e}} \Omega_\mathrm{e}.$$

The condition on grid conformity, however, is not required by the DG method; we only impose this condition to simplify the discussion.

To perform differentiation and integration operations, we introduce the nonsingular mapping $\boldsymbol{x} = \Psi(\boldsymbol{\xi})$ which defines a transformation from the physical Cartesian coordinate system $\boldsymbol{x} = (x, y)^\mathcal{T}$ to the local reference coordinate system $\boldsymbol{\xi} = (\xi, \eta)^\mathcal{T}$ defined on the reference triangle $\Omega_e = \{(\xi, \eta),\ -1 \le \xi, \eta \le 1,\ \xi + \eta \le 0, \}$.

Let us now represent the local element-wise solution $\boldsymbol{q}$ by an Nth order polynomial in $\boldsymbol{\xi}$ as

$$\boldsymbol{q}_N(\boldsymbol{\xi}) = \sum_{i=1}^{M_N} \psi_i(\boldsymbol{\xi}) \boldsymbol{q}_N(\boldsymbol{\xi}_i) \tag{8}$$

where $\boldsymbol{\xi}_i$ represents $M_N = \frac{1}{2}(N+1)(N+2)$ interpolation points and $\psi_i(\boldsymbol{\xi})$ are the associated multivariate Lagrange polynomials. For the interpolation points $\boldsymbol{\xi}_i$ we choose the nodal sets

based on the electrostatics [21] and Fekete [36] points; for simplicity we shall refer to these nodal sets collectively as Fekete points. We have already described the construction of the nodal basis functions in [18], [20] and, for the sake of brevity, omit this discussion here.

3.2. Integration

3.2.1. Area Integrals In order to complete the discussion of the local element-wise operations required to construct discrete spatial operators we must describe the integration procedure required by the integral formulation of all Galerkin methods. For any two functions f and g the 2D (area) integration $\mathcal{I}_A$ proceeds as follows

$$\mathcal{I}_A[f,g] = \int_{\Omega_e} f(\boldsymbol{x})\, g(\boldsymbol{x}) d\boldsymbol{x} = \sum_{i=1}^{M_Q} w_i^e \mid J^e(\boldsymbol{\xi}_i) \mid f(\boldsymbol{\xi_i})\, g(\boldsymbol{\xi_i})$$

where M_Q is a function of Q which represents the order of the cubature approximation. For w_i and $\boldsymbol{\xi}_i$ we use the high-order cubature rules for the triangle given in [35, 6, 27, 7]; because we use order $2N$ integration, which is exact for this equation set, then neither spatial filters nor smoothing diffusion operators are used. Furthermore, we omit the use of slope limiters.

3.2.2. Boundary Integrals The DG method also requires the evaluation of boundary integrals which is the mechanism by which the fluxes across element edges are evaluated and allows the discontinuous elements to communicate. For any two functions f and g the 1D (boundary) integration $\mathcal{I}_B$ proceeds as follows

$$\mathcal{I}_B[f,g] = \int_{\Gamma_e} f(\boldsymbol{x})\, g(\boldsymbol{x}) d\boldsymbol{x} = \sum_{i=0}^{Q} w_i^s \mid J^s(\boldsymbol{\xi}_i) \mid f(\boldsymbol{\xi_i})\, g(\boldsymbol{\xi_i})$$

where Q represents the order of the quadrature approximation. Using Legendre-Gauss quadrature we can use $Q = N$ to achieve order $2N$ accuracy.

3.3. Tangent and Normal Vectors of the Element Edges

Below it will become evident that in order to construct a discontinuous Galerkin discretization requires knowing a bit about the element geometry. In continuous Galerkin methods, such as the finite element method, the only required information is the basis functions, metric terms, and cubature rules. The DG method requires all of this finite element-type information plus some finite volume-type information regarding the element edges and the element neighbors sharing these edges. However, the good news for the DG method is that regardless of the order of the basis function, N, each element only has three edge neighbors (this is true only for conforming grids). This is the process by which a DG element shares its local information with its neighbors.

3.4. Semi-Discrete Equations

Applying the discontinuous Galerkin discretization to Eq. (1), and using Green's theorem yields the classical DG which we refer to as the *weak* form

$$\int_{\Omega_e} \left(\frac{\partial \boldsymbol{q}_N^{(e)}}{\partial t} - \boldsymbol{F}_N^{(e)} \cdot \boldsymbol{\nabla} - S_N^{(e)} \right) \psi_i(\boldsymbol{x}) d\boldsymbol{x} = -\sum_{l=1}^{3} \int_{\Gamma_e} \psi_i(\boldsymbol{x})\, \boldsymbol{n}^{(e,l)} \cdot \boldsymbol{F}_N^{(*,l)} d\boldsymbol{x} \tag{9}$$

where $F_N = F(\boldsymbol{q}_N)$ and $S_N = S(\boldsymbol{q}_N)$ with $\boldsymbol{F}$ and S given by Eqs. (2) and (3), respectively. Note that Eq. (9) states that $\boldsymbol{q}_N$ satisfies the equation on each element Ω_e for all $\psi \in \mathcal{S}$ where $\mathcal{S}$ is the finite-dimensional space

$$\mathcal{S} = \{\psi \in \mathcal{L}_2(\Omega) : \psi|_{\Omega_e} \in P_N(\Omega_e) \forall \Omega_e\},$$

P_N is the polynomial space defined on Ω_e and the union of these elements defines the entire global domain - that is, $\Omega = \bigcup_{e=1}^{N_e} \Omega_e$ with N_e representing the total number of triangular elements. It should be mentioned that in DG methods, the space $P_N - P_N$ can be used without violating the inf-sup (Ladyzhenskaya-Babuska-Brezzi) condition which must be observed by continuous Galerkin methods (such as the finite element method) in order to avoid the effects of spurious pressure modes.

In the boundary integral of Eq. (9) $\boldsymbol{n}$ is the outward pointing unit normal vector of the element edge Γ_e and $\boldsymbol{F}_N^*$ is the Rusanov numerical flux (although other fluxes are also possible)

$$\boldsymbol{F}_N^{(*,l)} = \frac{1}{2}\left[\boldsymbol{F}_N\left(\boldsymbol{q}_N^{(e)}\right) + \boldsymbol{F}_N\left(\boldsymbol{q}_N^{(l)}\right) - |\lambda^{(l)}|\left(\boldsymbol{q}_N^{(l)} - \boldsymbol{q}_N^{(l)}\right)\boldsymbol{n}^{(e,l)}\right] \tag{10}$$

where $\lambda^{(l)} = \max\left(|U^{(e)}| + \sqrt{\phi^{(e)}}, |U^{(l)}| + \sqrt{\phi^{(l)}}\right)$ with $U^{(e,l)} = \boldsymbol{u}^{(e,l)} \cdot \boldsymbol{n}^{(l)}$ being the normal component of velocity with respect to the edge Γ_e, and the superscripts e and l represent the element e and its three edge neighbors l. The normal vector $\boldsymbol{n}^{(e,l)}$ is defined as pointing outward from the element e to its edge neighbor l.

Integrating Eq. (9) by parts once more yields the *strong* form

$$\int_{\Omega_e} \psi_i(\boldsymbol{x}) \left(\frac{\partial \boldsymbol{q}_N^{(e)}}{\partial t} + \boldsymbol{\nabla} \cdot \boldsymbol{F}_N^{(e)} - S_N^{(e)}\right) d\boldsymbol{x} = \sum_{l=1}^{3} \int_{\Gamma_e} \psi_i(\boldsymbol{x})\, \boldsymbol{n}^{(e,l)} \cdot \left(\boldsymbol{F}_N^{(e)} - \boldsymbol{F}_N^{(*,l)}\right) d\boldsymbol{x} \tag{11}$$

which, although mathematically equivalent to the weak form, yields different numerical solutions. Based on previous studies (see Giraldo [18] and Kopriva [24]) we use the strong form exclusively in this paper.

3.5. Matrix Form of the Semi-Discrete Equations

Substituting the polynomial approximation

$$\boldsymbol{q}_N = \sum_{i=1}^{M_N} \psi_i \boldsymbol{q}_i$$

into Eq. (11) we can now write the semi-discrete system as

$$\int_{\Omega_e} \psi_i \psi_j \, d\boldsymbol{x} \, \frac{\partial \boldsymbol{q}_j^{(e)}}{\partial t} + \int_{\Omega_e} \psi_i \boldsymbol{\nabla} \psi_j \, d\boldsymbol{x} \cdot \boldsymbol{F}_j^{(e)} - \int_{\Omega_e} \psi_i \psi_j \, d\boldsymbol{x} \, S_j^{(e)} = \sum_{l=1}^{3} \int_{\Gamma_e} \psi_i \psi_j \boldsymbol{n}^{(e,l)} \, d\boldsymbol{x} \cdot \left(\boldsymbol{F}^{(e)} - \boldsymbol{F}^{(*,l)}\right)_j . \tag{12}$$

Next, note that by defining the following element matrices

$$M_{ij}^{(e)} = \int_{\Omega_e} \psi_i \psi_j \, d\boldsymbol{x}, \;\; \boldsymbol{M}_{ij}^{(e,l)} = \int_{\Gamma_e} \psi_i \psi_j \boldsymbol{n}^{(e,l)} \, d\boldsymbol{x}, \;\; \boldsymbol{D}_{ij}^{(e)} = \int_{\Omega_e} \psi_i \boldsymbol{\nabla} \psi_j \, d\boldsymbol{x}$$

allows us to write Eq. (12) in the following matrix form

$$M_{ij}^{(e)} \frac{\partial \boldsymbol{q}_j^{(e)}}{\partial t} + \left(\boldsymbol{D}_{ij}^{(e)}\right)^{\mathcal{T}} \boldsymbol{F}_j^{(e)} - M_{ij}^{(e)} S_j^{(e)} = \sum_{l=1}^{3} \left(\boldsymbol{M}_{ij}^{(e,l)}\right)^{\mathcal{T}} \left(\boldsymbol{F}^{(e)} - \boldsymbol{F}^{(*,l)}\right)_j \tag{13}$$

where the superscript e denotes an element-based evaluation and l denotes edge-based evaluations. Next, using the approach described in [20] for eliminating the mass matrix, we write

$$\widehat{\boldsymbol{D}}^{(e)} = \left(M^{(e)}\right)^{-1} \boldsymbol{D}^{(e)}, \; \widehat{\boldsymbol{M}}^{(e,l)} = \left(M^{(e)}\right)^{-1} \boldsymbol{M}^{(e,l)}$$

which then allows us to write Eq. (13) as follows

$$\frac{\partial \boldsymbol{q}_i^{(e)}}{\partial t} + \left(\widehat{\boldsymbol{D}}_{ij}^{(e)}\right)^{\mathcal{T}} \boldsymbol{F}_j^{(e)} - S_i^{(e)} = \sum_{l=1}^{3} \left(\widehat{\boldsymbol{M}}_{ij}^{(e,l)}\right)^{\mathcal{T}} \left(\boldsymbol{F}^{(e)} - \boldsymbol{F}^{(*,l)}\right)_j . \tag{14}$$

Equation (14) is the form we shall use in the construction of the explicit and semi-implicit discretizations.

3.6. Boundary Conditions

In all the test cases we only consider no-flux boundary conditions; we will extend our model to more general boundary conditions in future work. The no-flux boundary conditions are enforced by virtue of the statement

$$\boldsymbol{n} \cdot \boldsymbol{u} = 0 \tag{15}$$

at the boundaries. Thus, we seek to eliminate the normal component of the velocity to the no-flux boundary without altering the tangential component. The tangent vector to a boundary is obtained by $\boldsymbol{t} = \boldsymbol{k} \times \boldsymbol{n}$ which is equal to $\boldsymbol{t} = -n_y \boldsymbol{i} + n_x \boldsymbol{j}$. Thus we solve the following 2x2 system:

$$\begin{pmatrix} n_x & n_y \\ -n_y & n_x \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 0 \\ u_T \end{pmatrix} \tag{16}$$

where $u_T = \boldsymbol{t} \cdot \boldsymbol{u}$ is the tangential component of velocity. This boundary condition is imposed only weakly through the boundary integrals in Eq. (14); that is, it only comes in through the Rusanov flux.

4. Time-Integrator

In Sec. 3 we described the approximation of the spatial derivatives using the DG method. We are now in a position to describe the approximation of the time derivatives. Let us begin with the description of the explicit time-integration followed by the semi-implicit time-integration methods. We use a method of lines approach for both methods.

4.1. Explicit Method

In order to advance the solution in time while retaining some level of high-order accuracy we use third order strong stability preserving (SSP) Runge-Kutta methods (see [5] and [33]).

For completeness we define them now. Let us write the semi-discrete (in space) equations as follows

$$\frac{\partial \boldsymbol{q}}{\partial t} = R(\boldsymbol{q})$$

where

$$R(\boldsymbol{q}) = -\left(\widehat{\boldsymbol{D}}_{ij}^{(e)}\right)^{\mathcal{T}} \boldsymbol{F}_j^{(e)}(\boldsymbol{q}) + S_i^{(e)} + \sum_{l=1}^{3}\left(\widehat{\boldsymbol{M}}_{ij}^{(e,l)}\right)^{\mathcal{T}}\left(\boldsymbol{F}^{(e)}(\boldsymbol{q}) - \boldsymbol{F}^{(*,l)}(\boldsymbol{q})\right)_j \qquad (17)$$

and $\boldsymbol{q}$ is in fact $\boldsymbol{q}^{(e)}$, that is, the solution within each element e.

The SSP temporal discretization of this semi-discrete equation is

$$\begin{aligned} \text{for } k &= 1, ..., S: \\ \boldsymbol{q}^k &= \alpha_0^k \boldsymbol{q}^n + \alpha_1^k \boldsymbol{q}^{k-1} + \alpha_2^k \boldsymbol{q}^{k-3} + \beta^k \Delta t R(\boldsymbol{q}^{k-1}) \end{aligned}$$

where $\boldsymbol{q}^0 = \boldsymbol{q}^n$, $\boldsymbol{q}^S = \boldsymbol{q}^{n+1}$, S denotes the number of RK stages, and the coefficients α and β are given in Table I.

Method	k	α_0	α_1	α_2	β
RK3	1	1	0	0	1
	2	3/4	1/4	0	1/4
	3	1/3	2/3	0	2/3
RK34	1	1	0	0	1
	2	0	1	0	1/2
	3	2/3	1/3	0	1/6
	4	0	1	0	1/2
RK35	1	1	0	0	0.377268915331368
	2	0	1	0	0.377268915331368
	3	0.355909775063327	0.644090224936674	0	0.242995220537396
	4	0.367933791638137	0.632066208361863	0	0.238458932846290
	5	0	0.762406163401431	0.237593836598569	0.287632146308408

Table I. Coefficients for the explicit strong stability preserving third order Runge-Kutta methods.

In Fig. 1, we show the stability regions of these methods for the equation

$$\frac{dq}{dt} = \lambda q$$

which is a proxy for an advection-dominated equation; a reasonable approximation for the shallow water equations. Figure 1 shows that the stability region of RK35 is larger than those for RK34 and RK3. We are interested specifically in the region along the imaginary axis because this is the region most important to advection-dominated problems. In [8] all three RK methods were studied for the Navier-Stokes equations and it was determined that RK35 is indeed the most efficient of the third order methods. We have found similar results here for the shallow water equations. Based on the results of these studies, we use RK35 for the comparisons with the semi-implicit methods which we now describe.

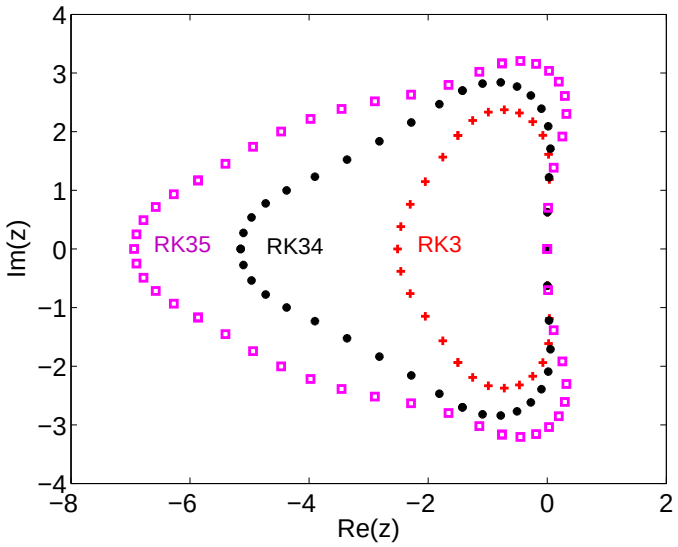


Figure 1. The stability regions for the explicit strongly stability preserving third order Runge-Kutta methods, RK3 (three stage), RK34 (four stage), and RK35 (five stage).

4.2. Semi-Implicit Method

To extend the size of the time-step we use a generalized semi-implicit method of order K. Let us write Eqs. (1)-(3) in the following compact vector form

$$\frac{\partial \boldsymbol{q}}{\partial t} = R(\boldsymbol{q}) \tag{18}$$

where $\boldsymbol{q} = (\phi_s, \boldsymbol{U}^{\mathcal{T}})^{\mathcal{T}}$ and $R(\boldsymbol{q})$ is defined in Eq. (17). This system can be represented by the equivalent form

$$\frac{\partial \boldsymbol{q}}{\partial t} = \{R(\boldsymbol{q}) - \delta_{SI} L(\boldsymbol{q})\} + [\delta_{SI} L(\boldsymbol{q})] \tag{19}$$

where $L(\boldsymbol{q})$ is the linear approximation to R given in Eq. (7) and contains the gravity wave terms (i.e., the fastest waves in this system, at least for subcritical flow). In Eq. (19) the curly brackets denote explicit time-integration while the square brackets denote implicit time-integration. Note that the variable δ_{SI} is a switch that yields a fully explicit method for $\delta_{SI} = 0$ and the semi-implicit method for $\delta_{SI} = 1$.

As was done in [17, 19] we now write the time-discretization in the general form:

$$\boldsymbol{q}^{n+1} = \sum_{k=0}^{K} \alpha_k \boldsymbol{q}^{n-k} + \gamma \Delta t \sum_{k=0}^{K} \beta_k [R(\boldsymbol{q}^{n-k}) - \delta_{SI} L(\boldsymbol{q}^{n-k})] + \gamma \Delta t \delta_{SI} L(\boldsymbol{q}^{n+1}) \tag{20}$$

where K denotes the order of the time-integrator. To simplify the discussion of the semi-implicit

formulation, let us now introduce the following auxiliary variables

$$\boldsymbol{q}_{tt} = \boldsymbol{q}^{n+1} - \sum_{k=0}^{K} \beta_k \boldsymbol{q}^{n-k}, \tag{21}$$

$$\boldsymbol{q}^E = \sum_{k=0}^{K} \alpha_k \boldsymbol{q}^{n-k} + \gamma \Delta t \sum_{k=0}^{K} \beta_k R(\boldsymbol{q}^{n-k}), \tag{22}$$

$$\widehat{\boldsymbol{q}} = \boldsymbol{q}^E - \sum_{k=0}^{K} \alpha_k \boldsymbol{q}^{n-k} \tag{23}$$

which then allows us to write Eq. (20) as

$$\boldsymbol{q}_{tt} = \widehat{\boldsymbol{q}} + \lambda L(\boldsymbol{q}_{tt}) \tag{24}$$

where $\lambda = \gamma \Delta t \delta_{SI}$. In Table II we list the coefficients for the backward difference formulas of order $K = 1, ..., 6$. In Figs. 2a and 2b we show the stability regions of the explicit and implicit

	K=1	K=2	K=3	K=4	K=5	K=6
α_0	1	4/3	18/11	48/25	300/137	360/147
α_1	0	-1/3	-9/11	-36/25	-300/137	-450/147
α_2	0	0	2/11	16/25	200/137	400/147
α_3	0	0	0	-3/25	-75/137	-225/147
α_4	0	0	0	0	12/137	72/147
α_5	0	0	0	0	0	-10/147
γ	1	2/3	6/11	12/25	60/137	60/147
β_0	1	2	3	4	5	6
β_1	0	-1	-3	-6	-10	-15
β_2	0	0	1	4	10	20
β_3	0	0	0	-1	-5	-15
β_4	0	0	0	0	1	6
β_5	0	0	0	0	0	-1

Table II. Coefficients for the backward difference formulas of orders $K = 1, ..., 6$.

BDF methods. In Fig. 2a the closed loops are the stability regions of the explicit BDF methods while in Fig. 2b the closed loops are the regions of instability of the implicit BDF methods. For the shallow water equations, we are interested in the region near the imaginary axis (for $Re(z) = 0$).

Let us now describe the semi-implicit method in terms of the governing equations. Note that the operator R referenced above is the same operator described for the explicit method. However, let us now write the full expression given in Eq. (24) in terms of the operator L. Substituting the linear operator defined in Eq. (7) into Eq. (24) results in the following system

$$\begin{aligned} \phi_{tt} &= \widehat{\phi} - \lambda \boldsymbol{\nabla} \cdot \boldsymbol{U}_{tt} \\ \boldsymbol{U}_{tt} &= \widehat{\boldsymbol{U}} - \lambda \left[\boldsymbol{\nabla} \cdot (\phi_{tt} \phi_B \boldsymbol{\mathcal{I}}_2) + f(\boldsymbol{k} \times \boldsymbol{U}_{tt}) - \phi_{tt} \boldsymbol{\nabla} \phi_B + \gamma \boldsymbol{U}_{tt} \right] \end{aligned} \tag{25}$$

where we have retained the continuous spatial operators for clarity. At this point, there is no difference between the semi-implicit formulation of a continuous Galerkin (e.g., finite elements)

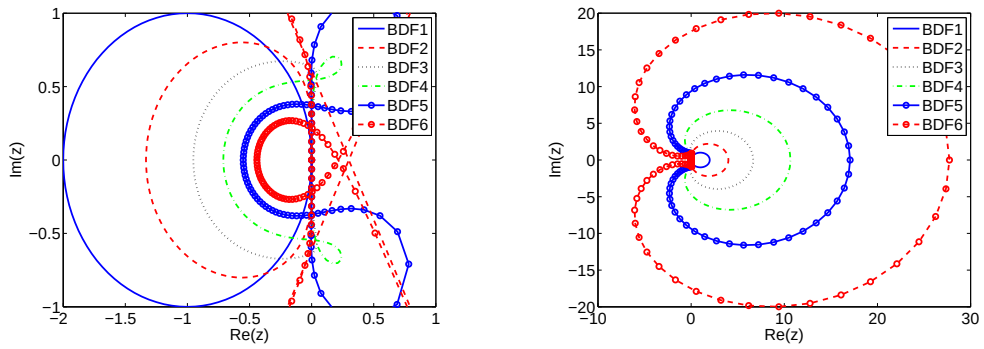


Figure 2. The stability regions of the a) explicit and b) implicit backward difference formulas of order $K = 1, ..., 6$.

and a discontinuous Galerkin method. The differences arise through the method selected for the discretization of the spatial operators. Replacing the continuous spatial operators with the DG discrete representations yields

$$\phi_{tt}^{(e)} = \widehat{\phi}^{(e)} - \lambda \left[\left(\widehat{\boldsymbol{D}}^{(e)} \right)^{\mathcal{T}} \boldsymbol{U}_{tt}^{(e)} - \sum_{l=1}^{3} \left(\widehat{\boldsymbol{M}}^{(e,l)} \right)^{\mathcal{T}} \left(\boldsymbol{U}_{tt}^{(e)} - \boldsymbol{U}_{tt}^{(*,l)} \right) \right] \tag{26}$$

$$\begin{aligned} \boldsymbol{U}_{tt}^{(e)} &= \widehat{\boldsymbol{U}}^{(e)} - \lambda \left(\widehat{\boldsymbol{D}}^{(e)} \right)^{\mathcal{T}} \left(\phi_{tt} \phi_B \boldsymbol{\mathcal{I}}_2 \right)^{(e)} \\ &- \lambda \sum_{l=1}^{3} \left(\widehat{\boldsymbol{M}}^{(e,l)} \right)^{\mathcal{T}} \left[\left(\phi_{tt} \phi_B \boldsymbol{\mathcal{I}}_2 \right)^{(e)} - \left(\phi_{tt} \phi_B \boldsymbol{\mathcal{I}}_2 \right)^{(*,l)} \right] \\ &- \lambda \left[f(\boldsymbol{k} \times \boldsymbol{U}_{tt}^{(e)}) - \phi_{tt}^{(e)} \boldsymbol{\nabla} \phi_B + \gamma \boldsymbol{U}_{tt}^{(e)} \right] \end{aligned} \tag{27}$$

where the flux values are specifically defined as

$$\boldsymbol{U}_{tt}^{(*,l)} = \frac{1}{2} \left[\boldsymbol{U}_{tt}^{(e)} + \boldsymbol{U}_{tt}^{(l)} - |\lambda_L| \boldsymbol{n}^{(l)} \left(\phi_{tt}^{(l)} - \phi_{tt}^{(e)} \right) \right] \tag{28}$$

and

$$\left(\phi_{tt} \phi_B \boldsymbol{\mathcal{I}}_2 \right)_{tt}^{(*,l)} = \frac{1}{2} \left[\left(\phi_{tt} \phi_B \boldsymbol{\mathcal{I}}_2 \right)^{(e)} + \left(\phi_{tt} \phi_B \boldsymbol{\mathcal{I}}_2 \right)^{(l)} - |\lambda_L| \boldsymbol{n}^{(l)} \left(\boldsymbol{U}_{tt}^{(l)} - \boldsymbol{U}_{tt}^{(e)} \right) \right]. \tag{29}$$

These equations can now be solved as a coupled system of linear equations for ϕ_{tt} and $\boldsymbol{U}_{tt}$. We use a GMRES solver with Jacobi preconditioning to solve this system. While this choice of preconditioner is not optimal, the resulting iterative method is nonetheless robust and efficient in terms of computational time and memory requirements since no global matrix problem ever needs to be stored.

In standard semi-implicit methods (e.g., see [16], [17], [19]), upon writing the fully discrete system the goal is then to apply a block LU decomposition in order to reduce the vector

system of equations into an equivalent scalar system of equations; for first order systems of equations, the resulting problem is in fact quite similar to a Helmholtz equation. Constructing the *Helmholtz problem* for general DG polynomial spaces and boundary conditions remains an open problem. Thus far, we only know how to construct the Helmholtz problem for collocated DG formulations (where the interpolation points coincide with the integration points) for a specific class of boundary conditions (see [29] and [30] for the solution to this problem for the Navier-Stokes equations).

A few additional comments regarding the semi-implicit discretization are in order. Since only the gravity waves (pressure gradient) are discretized implicitly, with the Rossby waves (advection operator) discretized explicitly, the model will be unconditionally stable with respect to the gravity waves for any time-step size as long as the conditional stability with respect to the Rossby waves is satisfied by the explicit methods. This is the reason why we show both the explicit and implicit stability regions of the BDF methods in Figs. 2a and 2b. Of course one could also choose to discretize all of the terms implicitly (including the nonlinear advection operator) - this is known as the fully-implicit method. The reason for choosing the semi-implicit method over the fully-implicit method is that in doing so we only need to contend with a linear matrix problem; for the fully-implicit method, we would have to solve a nonlinear matrix problem, which while not impossible, requires many more iterations for convergence (outer Newton loops, plus inner Krylov loops, [23]). For the types of shallow water problems that we are considering, the semi-implicit method should be more efficient than the fully-implicit method.

5. Numerical Experiments

For the numerical experiments, we use the normalized L^2 error norm

$$\|h_S\|_{L^2} = \sqrt{\frac{\int_\Omega (\phi_{\text{exact}} - \phi)^2 d\Omega}{\int_\Omega \phi_{\text{exact}}^2 \, d\Omega}}$$

computed at the cubature points to judge the accuracy of the methods. To compute the Courant number the elements are decomposed into their high-order (HO) grid points (which are in fact the Fekete points) and these grid points form a fine grid which we refer to as the HO cells. The velocities and grid spacings are then defined at the centers of these cells. Using these definitions the Courant number is then defined as

$$\text{Courant number} = \max \left(\frac{C\Delta t}{\Delta s} \right)_{HO}^{e} \quad \forall e \in [1, ..., N_e] \tag{30}$$

where $C = U + \sqrt{\phi}$ is the characteristic speed, $U = \sqrt{\boldsymbol{u} \cdot \boldsymbol{u}}$ is the magnitude of the velocity, and $\Delta s = \sqrt{\Delta x^2 + \Delta y^2}$ is the grid spacing. In addition, note that the Courant number based on the advection is given by Eq. (30) with $C = U$.

We use the symbol n_r to refer to the refinement level of the grid. This variable n_r represents the number of quadrilateral subdivisions in each of the Cartesian directions. For example, $n_r = 1$ corresponds to n_r^2 quadrilaterals and $2n_r^2$ triangles; the factor of 2 is required since each quadrilateral is subdivided into 2 triangles. Examples of square domains with $n_r = 1$, $n_r = 2$, and $n_r = 4$ are shown in Fig. 3.

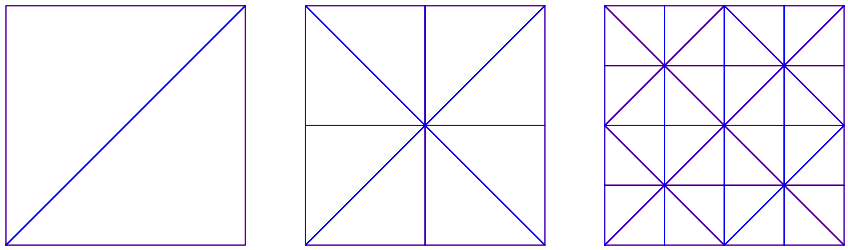


Figure 3. Grid refinement for the structured triangular grids with a) $n_r = 1$, b) $n_r = 2$, and c) $n_r = 4$.

5.1. Description of the Test Cases

We now describe the test cases and their solutions. It should be noted that all the tests presented below require no-flux boundary conditions at all four walls.

5.1.1. Linear Standing Wave This problem involves the transient solution of a linear inviscid standing wave without rotation which *sloshes* within a square basin of unit depth. From [22] we take the analytic solution as

$$\begin{aligned} h(\boldsymbol{x},t) &= \cos\pi x \cos\pi y \cos\sqrt{2}\pi t \\ u(\boldsymbol{x},t) &= \frac{1}{\sqrt{2}} \sin\pi x \cos\pi y \sin\sqrt{2}\pi t \\ v(\boldsymbol{x},t) &= \frac{1}{\sqrt{2}} \cos\pi x \sin\pi y \sin\sqrt{2}\pi t \end{aligned}$$

with $(x, y) \in [0, 1]^2$ and $t \in [0, 2]$. The source function, S, in Eq. (1) is zero and the flux tensor is linearized.

5.1.2. Linear Kelvin Wave This problem involves the transient solution of the linearized inviscid equations with rotation. From [13] we use the analytic solution

$$\begin{aligned} h(\boldsymbol{x},t) &= 1 + \exp\left(-\frac{y^2}{2}\right) \exp\left(-\frac{(x+5-t)^2}{2}\right) \\ u(\boldsymbol{x},t) &= \exp\left(-\frac{y^2}{2}\right) \exp\left(-\frac{(x+5-t)^2}{2}\right) \\ v(\boldsymbol{x},t) &= 0 \end{aligned}$$

with $f_0 = 0$, $\beta = 1$, and $(x, y) \in [-10, 10] \times [-5, 5]$ and $t \in [0, 10]$.

5.1.3. Nonlinear Rossby Soliton Wave This problem describes an equatorially trapped Rossby soliton wave [4]. The soliton wave starts off in the center of the domain. It then moves westward along the equator without changing shape. The asymptotically derived analytic

solution is given by

$$\begin{aligned} h(x,y,t) &= h^{(0)} + h^{(1)} \\ u(x,y,t) &= u^{(0)} + u^{(1)} \\ v(x,y,t) &= v^{(0)} + v^{(1)} \end{aligned}$$

where the superscripts (0) and (1) denote the zeroth and first order asymptotic solutions of the shallow water equations, respectively. They are given by

$$\begin{aligned} h^{(0)} &= \eta \left(\frac{-9+6y^2}{4} \right) e^{-\frac{y^2}{2}} \\ u^{(0)} &= \frac{\partial \eta}{\partial \xi} (2y)\, e^{-\frac{y^2}{2}} \\ v^{(0)} &= \eta \left(\frac{3+6y^2}{4} \right) e^{-\frac{y^2}{2}} \end{aligned}$$

and

$$\begin{aligned} h^{(1)} &= c^{(1)} \eta \frac{9}{16} \left(-5+2y^2 \right) e^{-\frac{y^2}{2}} + \eta^2 \Phi^{(1)}(y) \\ u^{(1)} &= c^{(1)} \eta \frac{9}{16} \left(3+2y^2 \right) e^{-\frac{y^2}{2}} + \eta^2 U^{(1)}(y) \\ v^{(1)} &= \frac{\partial \eta}{\partial \xi} \eta V^{(1)}(y) \end{aligned}$$

where $\eta(\xi,t) = A \operatorname{sech}^2 B\xi$, $\xi = x - c\,t$, $A = 0.771\,B^2$, $B = 0.394$, and $c = c^{(0)} + c^{(1)}$ where $c^{(0)} = -\frac{1}{3}$ and $c^{(1)} = -0.395\,B^2$. The variable η is the solution to the equation

$$\frac{\partial \eta}{\partial \tau} + \alpha_n \eta \frac{\partial \eta}{\partial \xi} + \beta_n \frac{\partial^3 \eta}{\partial \xi^3} = 0$$

which is a simplified form of the shallow water equations upon using the method of multiple scales presented in [3]. Finally, the remaining terms are given by

$$\begin{pmatrix} \Phi^{(1)}(y) \\ U^{(1)}(y) \\ V^{(1)}(y) \end{pmatrix} = e^{-\frac{y^2}{2}} \sum_{n=0}^{\infty} \begin{pmatrix} \varphi_n \\ u_n \\ v_n \end{pmatrix} H_n(y)$$

where $H_n(y)$ are the Hermite polynomials and φ_n, u_n, v_n are the Hermite series coefficients given in [4]. The Coriolis parameter is given by $f(y) = y$ where $(x,y) \in [-24,+24] \times [-8,+8]$ $t \in [0,10]$ and $g = 1$.

We include this analytic solution for completeness but one cannot use this test for validating the exponential convergence of the methods because the analytic solution is only a first order approximation. However, this solution can be used to check the phase speed of the soliton wave simulated by the numerical model.

5.1.4. Linear Stommel Problem The linear Stommel problem [34] is the exact steady-state solution of the linearized inviscid equations with rotation, wind stress, and bottom friction.

The analytic solution of this problem can be obtained by considering the linearized momentum equation as follows

$$\boldsymbol{\nabla}\phi + f(\boldsymbol{k}\times\boldsymbol{u}) + \gamma\boldsymbol{u} = \boldsymbol{\tau}.$$

If we define the Coriolis term as $f(y) = f_0 + \beta\left(y - \frac{L}{2}\right)$ and the wind stress as

$$\boldsymbol{\tau} = -\frac{\tau_0}{\rho H}\cos\left(\frac{\pi y}{L}\right)\boldsymbol{i} + 0\boldsymbol{j}$$

then taking the curl yields

$$\gamma\boldsymbol{\nabla}^2\psi + \beta\frac{\partial\psi}{\partial x} = -\frac{\tau_0\pi}{\rho H L}\sin\left(\frac{\pi y}{L}\right)$$

where we have written the velocity field in terms of the streamfunction as $u = -\frac{\partial\psi}{\partial y}$ and $v = \frac{\partial\psi}{\partial x}$. Assuming a separation of variables solution of the type $\psi(x,y) = \widehat{\psi}(x)\sin\left(\frac{\pi y}{L}\right)$ yields the following second order ordinary differential equation (ODE) for x

$$\gamma\frac{\partial^2\widehat{\psi}}{\partial x^2} + \beta\frac{\partial\widehat{\psi}}{\partial x} - \gamma\left(\frac{\pi}{L}\right)^2\widehat{\psi} = -\left(\frac{\pi}{L}\right)\frac{\tau_0}{\rho H}.$$

This ODE tells us that we need to seek solutions of the type $\widehat{\psi}(x) = Ce^{\lambda x} + C_0$ which, after substituting into the ODE, gives the two roots

$$\lambda_{1,2} = \frac{-\frac{\beta}{\gamma} \pm \sqrt{\left(\frac{\beta}{\gamma}\right)^2 + 4\left(\frac{\pi}{L}\right)^2}}{2}$$

with $C_0 = \frac{\tau_0}{\gamma\rho H}\frac{L}{\pi}$. Imposing zero streamfunction boundary conditions at the domain boundaries yields the final solution of the streamfunction as

$$\psi(x,y) = \left(C_0 + C_1 e^{\lambda_1 x} + C_2 e^{\lambda_2 x}\right)\sin\left(\frac{\pi y}{L}\right)$$

where

$$C_1 = C_0\frac{1 - e^{\lambda_2 L}}{e^{\lambda_2 L} - e^{\lambda_1 L}} \qquad \text{and} \qquad C_2 = -C_0\frac{1 - e^{\lambda_1 L}}{e^{\lambda_2 L} - e^{\lambda_1 L}}.$$

This then yields the analytic solution

$$\begin{aligned}
h(\boldsymbol{x}) &= \frac{1}{g}\left[C_0\beta\frac{L}{\pi}\cos\left(\frac{\pi y}{L}\right) + f(y)\psi(x,y) + \frac{\gamma\pi}{L}\cos\left(\frac{\pi y}{L}\right)\left(\frac{C_1}{\lambda_1}e^{\lambda_1 x} + \frac{C_2}{\lambda_2}e^{\lambda_2 x}\right)\right] \\
u(\boldsymbol{x}) &= -\frac{\pi}{L}\widehat{\psi}(x)\cos\left(\frac{\pi y}{L}\right) \\
v(\boldsymbol{x}) &= \left(C_1\lambda_1 e^{\lambda_1 x} + C_2\lambda_2 e^{\lambda_2 x}\right)\sin\left(\frac{\pi y}{L}\right).
\end{aligned}$$

The constants required to completely define the solution are $f_0 = 1\times10^{-4}$, $\beta = 1\times10^{-11}$, $\gamma = 1\times10^{-6}$, $g = 10$, $\rho = 1000$, $\tau = 0.2$, $H = 1000$, and $L = 1\times10^{6}$. The models are integrated between 200 to 400 days in order to reach steady-state. We regard steady-state as the condition when the error norms cease to decrease.

5.1.5. Nonlinear Riemann Problem We used the nonlinear Riemann problem previously (see [20]) in order to validate the spatial operators of our DG model and its slope limiters. We follow the outline of the problem presented in Toro [37]. The source function S is set to zero; this leaves a balance between the time rate of change of the conservation variable $\boldsymbol{q}$ and the the divergence of the flux tensor. Following [37] we use

$$h(x,y,0) = \begin{cases} 2.5 & \text{if} \quad r \leq R \\ 0.5 & \text{if} \quad r > R \end{cases}$$

with $\boldsymbol{u}(x,y,0) = 0 \ \forall (x,y) \in [-20,20]^2$ where $r = \sqrt{x^2+y^2}$, $R = 2.5$, and $t \in [0, 0.4]$. The cylindrical wave is positioned initially at the origin and moves outward for increasing time t.

5.2. Comparison of the Explicit and Semi-Implicit Models

In the sections below, we compare the accuracy and efficiency of the explicit RK35 method with the semi-implicit BDF methods of order $K \leq 6$. For all simulations, the largest Courant number shown for RK35 represents the maximum Courant number allowed by this method. The smallest Courant number shown for the semi-implicit BDF methods represents the maximum Courant number allowed by the explicit BDF methods; the only exception is the time-step convergence study that we now describe.

5.2.1. Time-Step Convergence Study The first study we conduct is the rate of convergence of explicit and implicit time-integrators. For this study we use the linear standing wave problem with $n_r = 1$ refinement level, corresponding to two triangular elements, and 14th order ($N = 14$) polynomials. For this resolution, the best possible normalized h_S L^2 error norm that can be achieved by the model is 1×10^{-9} which we obtained experimentally as $\Delta t \to 0$; this we consider to be the *exact numerical solution*.

In Fig. 4 we show the results for the RK35 and BDF methods. The maximum time-step used for RK35 is the maximum allowed by the method. For the BDF methods of order $K \leq 4$, the smallest time-steps corespond to the explicit methods while the last few points correspond to the implicit methods. For the BDF methods of order $K \geq 5$, all the simulations are obtained with the implicit methods; because these methods are high-order in time, they achieve the exact numerical solution for time-steps much larger than those allowed by the explicit method.

We define the time rate of convergence as

$$\text{rate} = abs\left(\sum_{i=1}^{N_T} \frac{\log\left[\text{error}_{\Delta t_{i+1}} / \text{error}_{\Delta t_i}\right]}{\log\left[\Delta t_i / \Delta t_{i+1}\right]} \right)$$

where Δt_i are the N_T time-step values. Figure 4 shows that RK35 is indeed formally third order accurate, and that the BDF methods achieve their theoretical values of order K. Furthermore, the explicit and all implicit methods yield exact mass conservation (to within machine double precision) not just for this test case but for all cases discussed below. Let us now examine the accuracy and efficiency of the semi-implicit BDF time-integrators for various test cases and compare them to the explicit RK35 time-integrator.

5.2.2. Linear Standing Wave In Figs. 5a and 5b we show the L^2 error norms and the wallclock time as functions of Courant number for various time-integrators for the linear standing wave

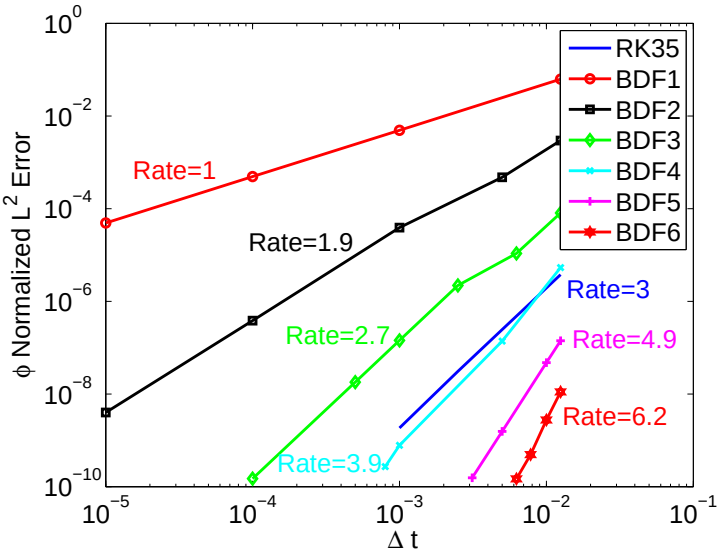
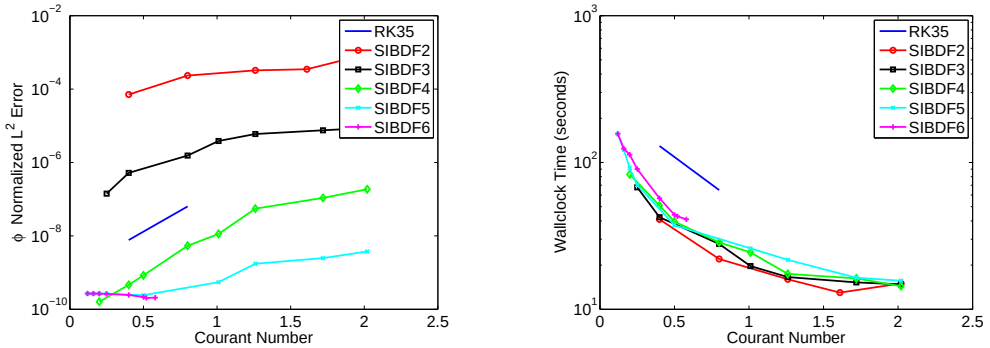


Figure 4. Linear Standing Wave. The normalized h_S L_2 error as a function of time-step for various time-integrators. All runs use $n_r = 1$ and $N = 14$.



a) b)

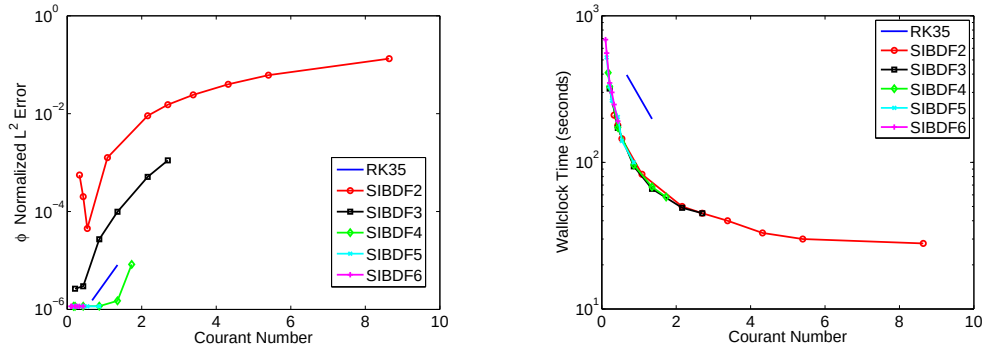
Figure 5. Linear Standing Wave. The a) normalized h_S L_2 error and b) wallclock time as functions of Courant number for various time-integrators. All runs use $n_r = 6$ and $N = 10$.

problem. In these simulations we use tenth order polynomials (N=10) with 72 triangular elements (corresponding to $n_r = 6$). Figure 5a shows that the RK35 explicit method is more accurate than the semi-implicit BDF methods of order $K \leq 3$. For $K \geq 4$ the BDF methods are more accurate. Figure 5b shows that all of the BDF methods are more efficient than the explicit RK35 method for the same Courant number. Furthermore, the BDF methods of order $K \leq 6$ allow larger Courant numbers than the explicit RK35 method and the lower order BDF

methods are more efficient than the high-order BDF methods.

The small Courant numbers reported for BDF6 in Fig. 5b needs to be explained. To understand these results, let us begin by discussing the stability regions of the implicit BDF methods in Fig. 2b. Along the imaginary axis all the BDF methods are stable for large z. Clearly, for some range of z, the BDF methods of orders $K \geq 3$ become unstable. This is observed in Fig. 5b for $Re(z) = 0$ and $|Im(z)| < 5$ for BDF4, for example. Once this first instability region is reached, we stop increasing the Courant numbers which results in the small Courant numbers reported in Figs. 5. Note that we do this for all of the simulations throughout this paper.

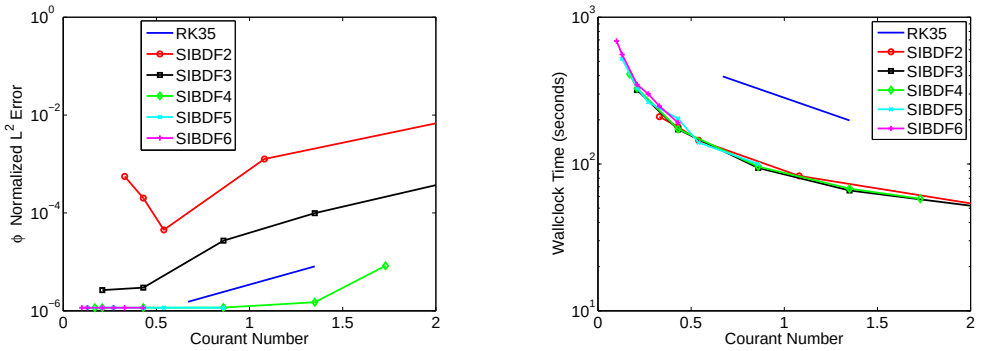
5.2.3. Linear Kelvin Wave In Figs. 6a and 6b we show the normalized h_S L^2 error norms and the wallclock time as functions of Courant number for various time-integration methods for the linear Kelvin wave problem. In these simulations we, once again, use tenth order polynomials (N=10) with 256 triangular elements (corresponding to $n_r^x = 16$ and $n_r^y = 8$). Figure 6a shows that the RK35 explicit method is more accurate than the semi-implicit BDF methods of order $K \leq 3$. For $K \geq 4$ the BDF methods are more accurate, as in the previous case. Figure 6b shows that all of the semi-implicit BDF methods are more efficient than the explicit RK35 method for the same Courant numbers. Once again, the BDF methods allow larger Courant numbers with BDF2 allowing a Courant number almost one order of magnitude larger than the RK35 method.



a) b)

Figure 6. Linear Kelvin Wave. The a) normalized h_S L_2 error and b) wallclock time as functions of Courant number for various time-integrators. All runs use $n_r^x = 16$, $n_r^y = 8$, and $N = 10$.

Because the BDF2 method allows such large Courant numbers, it makes it very difficult to discern the performance of the other BDF methods. In Figs. 7a and 7b we show the results using smaller Courant numbers. In these figures it becomes obvious that the best method is the BDF4 since it yields the best error norms (together with BDF5 and BDF6) while allowing for larger Courant numbers than RK35, BDF5, and BDF6 and thereby yielding the optimal result when considering both accuracy and efficiency.



a) b)

Figure 7. Linear Kelvin Wave. A close-up of the a) normalized h_S L_2 error and b) wallclock time as functions of Courant number for various time-integrators. All runs use $n_r^x = 16$, $n_r^y = 8$, and $N = 10$.

5.2.4. Rossby Soliton Wave Figure 8a shows the wallclock time as a function of Courant number for various time-integration methods for the nonlinear Rossby soliton wave problem. In Fig. 8b we show the same simulations but for smaller Courant numbers. Recall that for this test we only have a first-order solution which is not sufficiently accurate for performing a convergence study; however, we can use it to determine whether the solitons are moving with the proper wave speed. We consider a numerical solution to be accurate if it agrees exactly with the semi-analytic solution as to the position of the highest peak of the solitons.

In these simulations we use eighth order polynomials (N=8) with 384 triangular elements (corresponding to $n_r^x = 24$ and $n_r^y = 8$). Figure 8 shows that the semi-implicit BDF methods $K < 6$ are more efficient than the explicit RK35 method for the same Courant numbers. Additionally, the semi-implicit BDF methods $K \leq 3$ admit larger Courant numbers than the explicit RK35 method. Since this case is nonlinear, both the explicit and implicit BDF methods are used in tandem to solve the problem. Therefore in this test case, the stability regions of both the implicit and explicit BDF methods are relevant. Looking at the stability region of the explicit BDF methods given in Fig. 2a we note that the BDF5 and BDF6 have particularly small stability regions that result in the small Courant numbers reported in Fig. 8 for these methods. The BDF methods of order $K \leq 4$ have larger stability regions in both the explicit and implicit forms and is the reason why these methods perform more efficiently than the high-order BDF ($K \geq 5$) and RK35 methods.

5.2.5. Linear Stommel Problem Figure 9a shows the wallclock time as a function of Courant number for various time-integration methods for the linear Stommel problem; in Fig. 9b we show a close-up of the same simulations for smaller Courant numbers. For this case we have a steady-state analytic solution and so the accuracy of the time-integrator only plays a small role. The accuracy of the model is completely dependent on the polynomial order of the DG method; the only role that the time-integrator has is to maintain the stability of the solution while doing so as efficienctly as possible.

In these simulations we use eighth order polynomials (N=8) with 32 triangular elements

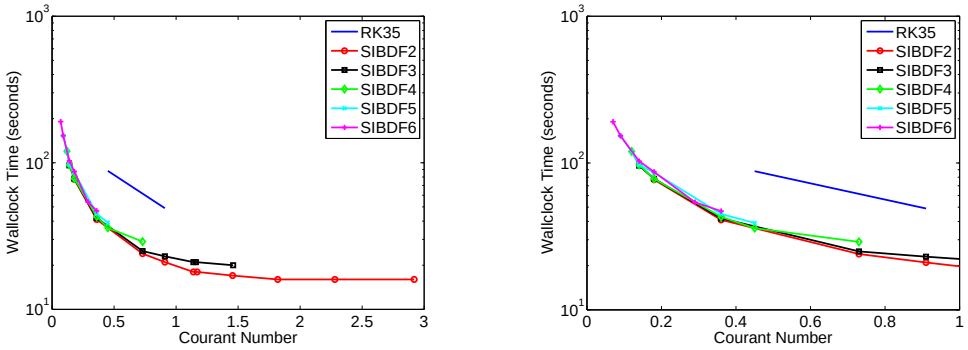


Figure 8. Rossby Soliton Wave. The wallclock time as a function of Courant number for various time-integrators. Figure a) shows the results for large Courant numbers while b) shows them for small Courant numbers. All runs use $n_r^x = 24$, $n_r^y = 8$, and $N = 8$.

(corresponding to $n_r = 4$). Figure 9a shows that the BDF methods $K \leq 2$ allow larger Courant numbers than the other methods. More importantly, Fig. 8b shows that all of the BDF methods are more efficient than the explicit RK35 method for the same Courant numbers. In addition, for this test case, the implicit BDF methods admit as large a Courant number as the explicit RK35 method. The fact that the implicit BDF methods are more efficient than the explicit RK35 even for the same Courant number is impressive especially since the implicit BDF methods require much more machinery to solve the problem. Recall that implicit/semi-implicit methods require the use of iterative solvers (in this case GMRES) and preconditioners (in this case Jacobi preconditioning) in order to solve the resulting linear matrix problem. Even with all of this machinery, the implicit methods are more efficient than an explicit RK method - this does not seem possible at first glance. The reason for these surprising results is simple: for Courant numbers less than 1 the implicit BDF methods require fewer than 5 GMRES iterations to converge; recall that the RK35 method requires 5 stages. Thus at this range of Courant numbers the implicit BDF methods are more efficient than RK35 with respect to operation count which translates to smaller wallclock times. For the larger Courant number values, the number of iterations are greater than 5 but the larger time-steps (hence fewer time-integration loops) compensate for the extra costs incurred with respect to operation count.

One final comment is in order. Since this test case is linear and the solution is steady-state, we could have used infinitely large Courant numbers for the implicit BDF methods $K \leq 2$. We have only chosen to report the maximum Courant numbers that maintained stability for the nonlinear Stommel problem. Since we do not have an analytic solution to the nonlinear Stommel problem, we then use the linear problem to ensure that we are achieving L^2 errors of 1×10^{-6} which is the *exact numerical solution* for eighth order polynomials with $n_r = 4$ (see [20]). This means that the results reported in Fig. 9 are also representative of the types of efficiency gains offered by the semi-implicit BDF methods for nonlinear problems. However, it should be noted that the reason why such large Courant numbers can be used in this test case

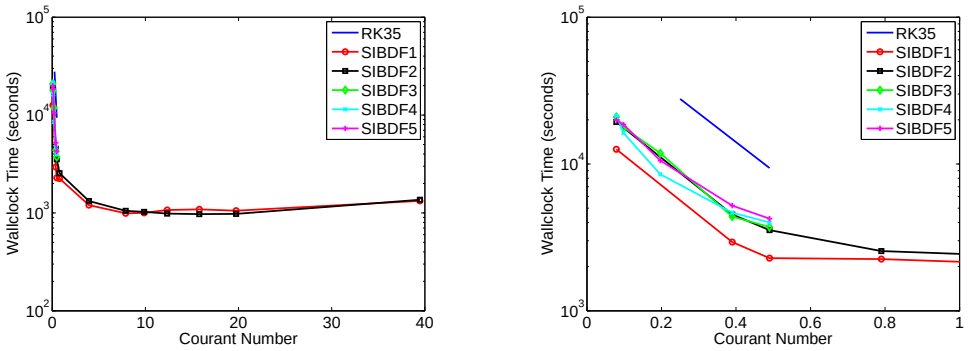


Figure 9. Linear Stommel Problem. The wallclock time as a function of Courant number for various time-integrators. Figure a) shows the results for large Courant numbers while b) shows them for small Courant numbers. All runs use $n_r = 4$ and $N = 8$.

has to do with the disparity between the speed of the gravity waves (the height of the fluid) and the Rossby waves. For the Stommel problem, the gravity waves are much faster than the Rossby waves and this will be the case for all deep ocean flows. Let us now discuss a type of flow problem for which the semi-implicit method is not well suited.

5.2.6. Nonlinear Riemann Problem In Figs. 10a and 10b we show the wallclock time as a function of Courant number for various time-integrators for the nonlinear Riemann problem. Note that we only report the explicit RK35 and explicit BDF methods. The results for this test show that the explicit BDF methods of order $K \leq 4$ compete with RK35 in terms of efficiency. The stability regions of the BDF methods of order $K \geq 5$ are too small and, while faster than RK35 for a given Courant number, cannot compete with the maximum Courant number admitted by RK35. Let us now discuss why we do not show results for the semi-implicit BDF methods.

We cannot use the semi-implicit BDF methods for this case because the Rossby waves are faster than the gravity waves. This means that the linearization used to construct the equations in (6) is no longer valid. The linearization used in the current semi-implicit formulation assumes that ϕ_B is much greater than ϕ_S which is not true for the Riemann problem (as is evident by the initial conditions where $\phi_B = 0.5$ and $max(\phi_S) = 2$ meters^2/second). We show the result of the Riemann problem only to point out the limitation of our current approach. Let us now discuss some possible solutions to this dilemma.

Defining the Froude number as the ratio of propagation speeds of Rossby (call them R) and gravity (call them G) waves, then if the flow is subcritical (i.e., $G > R$) then the fix to the problem is relatively simple. Instead of linearizing about a constant state, say ϕ_B, we linearize instead about the known state at the current time-step (i.e., $\phi_S^n + \phi_B$, where n denotes the current time-level). This presents very little changes to the current semi-implicit approach. On the other hand, if the flow is supercritical (i.e., $R > G$) then nothing in the semi-implicit machinery can improve the efficiency since the terms responsible for the fastest waves in the system are discretized explicitly in time.

The simplest solution, given the methodology described in this paper, is to switch from the semi-implicit to the explicit methods which is achieved by setting the parameter $\delta_{SI} = 0$ in the code - this, of course, has to be done with the additional constraint that the time-step be changed in order to satisfy the explicit stability region of the BDF methods.

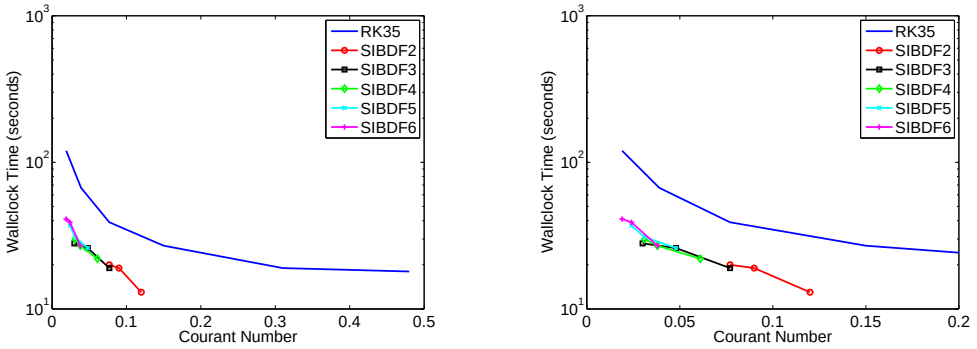


Figure 10. Riemann Problem. The wallclock time as a function of Courant number for various time-integrators. Figure a) shows the results for large Courant numbers while b) shows them for small Courant numbers. All runs use $n_r = 100$ and $N = 1$.

Another approach is to discretize the equations fully implicitly in time which then requires the solution of a nonlinear matrix problem. We prefer the first approach for its simplicity and to this end we are developing tools to automate the selection of the time-step as well as the value of the switch δ_{SI}. Our analysis has shown that not all K order methods are created equally. For example, taking all the results collectively shows that the RK35 method behaves most like the BDF4 method and so the optimal combination would be to use the semi-implicit BDF4 as long as the linearization is valid and then switching to the explicit RK35 when the linearization breaks down or supercritical flow is encountered. The value of such a hybrid solution strategy can be appreciated by considering the semi-implicit time-integration of a tsunami wave beginning in the middle of the deep ocean. As the wave approaches the coastline, the semi-implicit linearization breaks down and the flow becomes supercritical which then requires the code to switch to explicit mode. We hope to report the results of such simulations in the near future.

6. Conclusions

We present a high-order family of semi-implicit time-integration methods based on backward difference formulas (BDF) for the triangular discontinuous Galerkin method as applied to the oceanic shallow water equations; We use a high-order discontinous Galerkin method defined on unstructured triangular elements which is especially useful when attempting to resolve the complex geometry resulting from the representation of coastlines in coastal ocean models. In this work, we have extended the explicit in time high-order DG method that was shown to be exponentially convergent (for smooth problems) to semi-implicit in time. The semi-implicit

BDF time-integrators of order $K \geq 4$ are shown to yield better accuracy than the third order explicit Runge-Kutta method. Furthermore, the BDF methods of order $K \leq 4$ require far less wallclock time to deliver these solutions. We show that the semi-implicit BDF methods, even without any optimization and without the use of sophisticated preconditioners, yields better efficiency than the explicit RK method. We expect that on a parallel computer, with the aid of preconditioning and reduction of the implicit problem to a Helmholtz problem, the speed-up of the semi-implicit method compared to the fastest explicit methods will be further increased. In future work we plan on adding lateral diffusion, variable bathymetry, and wetting and drying algorithms to the model in order to perform tsunami, storm surge, and inundation simulations.

ACKNOWLEDGEMENTS

The first author (FXG) gratefully acknowledges the support of the Office of Naval Research through program element PE-0602435N and the Naval Postgraduate School through a research initiative grant.

REFERENCES

1. D. Alevras, *Simulations of the Indian Ocean tsunami with realistic bathymetry using a high-order triangular discontinuous Galerkin shallow water model*, Naval Postgraduate School, Masters Thesis (2008).
2. V. Aizinger and C. Dawson, A discontinuous Galerkin method for two-dimensional flow and transport in shallow water, *Advances in Water Resources* **25**, 67-84 (2002).
3. J.P. Boyd, Equatorial solitary waves. Part 1: Rossby solitons, *Journal of Physical Oceanography* **10**, 1699-1717 (1980).
4. J.P. Boyd, Equatorial solitary waves. Part 3: westward-travelling modons, *Journal of Physical Oceanography* **15**, 46-54 (1985).
5. B. Cockburn, and C-W. Shu, Runge-Kutta discontinuous Galerkin methods for convection-dominated problems, *Journal of Scientific Computing* **16**, 173-261 (2001).
6. R. Cools, and P. Rabinowitz, Monomial cubature rules since Stroud: A compilation, *Journal of Computational and Applied Math*, **48**, 309-326, (1993).
7. R. Cools, Monomial cubature rules since Stroud: A compilation - Part 2, *Journal of Computational and Applied Math*, **112**, 21-27, (1999).
8. T.J. De Luca, *Performance of hybrid Eulerian-Lagrangian semi-implicit time-integrators for nonhydrostatic mesoscale atmospheric modeling*, Naval Postgraduate School, Master's Thesis (2007).
9. V. Dolejsi, Semi-implicit interior penalty discontinuous Galerkin methods for viscous compressible flows, *Communications in Computational Physics* **4**, 231-274 (2008).
10. V. Dolejsi, M. Feistauer, J. Hozman, Analysis of semi-implicit DGFEM for nonlinear convection-diffusion problems on nonconforming meshes, *Computer Methods in Applied Mechanics and Engineering* **196**, 2813-2827 (2007).
11. V. Dolejsi and M. Feistauer, A semi-implicit discontinuous Galerkin finite element method for the numerical solution of inviscid compressible flow, *Journal of Computational Physics* **198**, 727-746 (2004).
12. F. Dupont and C.A. Lin, The adaptive spectral element method and comparisons with more traditional formulations for ocean modeling, *Journal of Atmospheric and Oceanic Technology* **21**, 135-147 (2004).
13. C. Eskilsson, and S.J. Sherwin, A triangular spectral/hp discontinuous Galerkin method for modelling 2D shallow water equations, *International Journal for Numerical Methods in Fluids* **45**, 605-623 (2004).
14. M. Feistauer and V. Kucera, On a robust discontinuous Galerkin technique for the solution of compressible flow *Journal of Computational Physics* **224**, 208-221 (2007).
15. M. Feistauer, V. Dolejsi, V. Kucera, On the discontinuous Galerkin method for the simulation of compressible flow with wide range of Mach numbers *Computing and Visualization in Science* **10**, 17-27 (2007).
16. F.X. Giraldo, J.B. Perot, and P.F. Fischer, A spectral element semi-Lagrangian (SESL) method for the spherical shallow water equations, *Journal of Computational Physics* **190**, 623-650 (2003).
17. F.X. Giraldo, Semi-implicit time-integrators for a scalable spectral element atmospheric model, *Quarterly Journal of the Royal Meteorological Society* **131**, 2431-2454 (2005).

18. F.X. Giraldo, High-order triangle-based discontinuous Galerkin methods for hyperbolic equations on a rotating sphere, *Journal of Computational Physics* **214**, 447-465 (2006).
19. F.X. Giraldo, Hybrid Eulerian-Lagrangian semi-implicit time-integrators, *Computers and Mathematics with Applications* **52**, 1325-1342 (2006).
20. F.X. Giraldo and T. Warburton, A high-order triangular discontinuous Galerkin oceanic shallow water model, *International Journal for Numerical Methods in Fluids* **56**, 899-925 (2008).
21. J.S. Hesthaven, From electrostatics to almost optimal nodal sets for polynomial interpolation in a simplex, *SIAM Journal on Numerical Analysis* **35**, 655-676 (1998).
22. M. Iskandarani, D.B. Haidvogel, and J.P. Boyd, A staggered spectral element model with application to the oceanic shallow water equations, *International Journal for Numerical Methods in Fluids* **20**, 393-414 (1995).
23. D.A. Knoll, D.E. Keyes, Jacobian-free Newton-Krylov methods: a survey of approaches and applications, *Journal of Computational Physics* **193**, 357-397 (2004).
24. D.A. Kopriva, Metric identities and the discontinuous spectral element method on curvilinear meshes, *Journal of Scientific Computing* **26**, 301-327 (2006).
25. E.J. Kubatko, J.J. Westerink, and C. Dawson, hp discontinuous Galerkin methods for advection dominated problems in shallow water flow, *Computer Methods in Applied Mechanics and Engineering* **196**, 437-451 (2006).
26. H. Li and R.X. Liu, The discontinuous Galerkin finite element method for the 2d shallow water equations, *Mathematics and Computers in Simulation* **56**, 171-184 (2001).
27. J. Lyness and R. Cools, A survey of numerical cubature over triangles, *Applied Mathematics* **48**, 127-150 (1994).
28. J.F. Remacle, S.S. Frazão, X.G. Li, and M.S. Shephard, An adaptive discretization of shallow-water equations based on discontinuous Galerkin methods, *International Journal for Numerical Methods in Fluids* **52**, 903-923 (2006).
29. M. Restelli, *PhD Dissertation Title*, PhD Thesis (2007).
30. M. Restelli and F.X. Giraldo, *SIAM Journal on Scientific Computing*, in review (2008).
31. A. Robert, J. Henderson, and C. Turnbull, An implicit time integration scheme for baroclinic models of the atmosphere, *Monthly Weather Review* **100**, 329-335 (1972).
32. D. Schwanenberg and J. Köngeter, A discontinuous Galerkin method for the shallow water equations with source terms, In *Discontinuous Galerkin Methods*, B. Cockburn, G.E. Karniadakis, C-W. Shu (eds.) Springer: Heidelberg, 289-309 (2000).
33. R.J. Spiteri and S.J. Ruuth, A new class of optimal high-order strong-stability-preserving time discretization methods, *SIAM Journal on Numerical Analysis* **40**, 469-491 (2002).
34. H. Stommel, The westward intensification of wind-driven ocean currents, *Transactions of the American Geophysics Union* **29** 202-206 (1948).
35. A.H. Stroud, Approximate Calculation of Multiple Integrals, Prentice-Hall Publishing, New Jersey, (1971).
36. M.A. Taylor, B.A. Wingate, and R.E. Vincent, An algorithm for computing Fekete points in the triangle, *SIAM Journal on Numerical Analysis* **38**, 1707-1720 (2000).
37. E. Toro, Shock-capturing methods for free-surface shallow flows, p. 245, Wiley, New York (2001).